[redacted]

DEPARTMENT OF EDUCATION - FSA

FMS MODERNIZATION PARTNER - PHASE IV

ACCOUNTS RECEIVABLE (AR) AND ACCOUNTS PAYABLE (AP) RELEASE 4.3 TEST PLAN

Document Author: Lynn Chang
Creation Date: 05/01/02 9:23 AM
Last Updated: 05/29/02 4:00 PM

Document Control

Reviewers

Name	Position
Paul Stonner	FMS Director
Todd Elliott	FMS Project Manager
Jennifer Alden	Implementation Manager
Christine Ponzi	Transition Manager

Distribution

Copy No.	Name	Location

TABLE OF CONTENTS

1. Design Testing Overview 4
2. Design Test Scope and Expectations 6
3. DT Test Schedule 7
4. DT Roles and Responsibilities 7
   - Test Coordinator 8
   - Program Workstream 8
   - Management Office 9
   - Integration Workstream 9
   - Technical Infrastructure 9
5. Executing and Validating Test Cases 10
6. Issue Tracking and Problem Diagnosis and Resolution 10
7. Test Tracking and Documentation 12
8. Test Data 12
9. Test Scripts 12
10. Test Cycles 13
11. Design Test Results 13
12. Client Acceptance Approach 13
13. EXHIBITS 14
    - Exhibit 1: Unit and System/Integration Sample Test Script Template 14
    - Exhibit 2 – Definitions of Test Case / Scenario elements 15
    - Exhibit 3 - Testing Incident Report 17
    - Exhibit 4. Readiness Check list 18
14. APPENDICES 19
    - Appendix 1: Test Cycle Time Allocation 19
    - Appendix 2: System and Integration Test Cycle Breakdown 20
    - Appendix 3: Detailed Testing Time Table 21

1. Design Testing Overview

Design Testing (DT) is the functional test of the system prior to production, and validates the system's design through the simulation of actual business processes. Business functions and system functions are executed in a controlled environment using predefined test scripts to ensure integrity and confidence in the results. A Design Test also incorporates the new business procedures to validate that the system properly enables their execution. The system's business events are examined from the user or business perspective, rather than attempting to test every possible condition from the design perspective.

Unit Testing allows for robust testing that exercises logical branches and conditions. This testing is conducted by both the developers and the functional resources on the team to ensure that all critical coding issues and business logic defects are addressed, documented, corrected and regression tested before Design Test begins. Unit Testing is performed independent of the Design Test and involves the following:

- Testing all custom-designed business events and conditions
- Documentation of the expected / actual results at the transaction level
- Documentation of the expected / actual results at the file level
- Documentation of the accounting treatment within FMS per file / per transaction
- Documentation of all test scripts (once they have passed)

Once Unit Testing is complete, Design Test begins. There are three FMS components in Design Test for this phase:

- System Test
- Integration Test
- User Acceptance Test

The goal and objectives of a System Test - in addition to validating the design - are to confirm set ups, validate configurations, and to identify either coding errors or errors in business logic within the FMS environment. This test will serve as an "end-to-end" test within the confines of FMS.

The goal and objectives of an Integration Test – in addition to further validating the design – are to ensure that FMS is able to receive data from LARS, and to validate accounting of the transactions received in FMS. The Integration Test also serves as a second mechanism to re-affirm that the objectives of System Test are met. The Integration Test will also confirm cross-functional handoff points and processes. The testing should also identify and resolve all go-live issues. The deployment planning

approach and assumptions should also be considered and captured during Integration Testing.

The goal and objectives of a User Acceptance Test – in addition to further validating the design – are to ensure that training material and methods are properly documented and communicated to all possible users, and to ensure that all business critical user requirements are met. Potential users are heavily involved at this point of testing and it is during User Acceptance Test when potential users have the ability to make final comments regarding the design functionality and appearance of the interfaces before the system is placed into production.

This document provides the details of the Design Test activities and work products. Samples of work products are included in the exhibit section of this document. All scripts and work products will be placed under configuration management. Any changes and updates to scripts will follow CM guidelines (please refer to the CM Plan located in: F:\Configuration Management\CM Plan).

Entry and Exit criteria have been established as management control gates for the design test process. Entry criteria have been captured and listed on the following checklist.

Design Test Workstream Readiness Checklists:

Prior to entry into a Design Test there are a number of activities and events that must take place and documents that should be completed. These events and documents have been captured in the DT workstream readiness checklist and will be reviewed at the readiness session by the management office. Each workstream will be required to complete all relevant sections and submit for review. A sample of this document is displayed in Exhibit 4. All sections of the checklist must be signed off as being successfully completed or signed off as incomplete with approval. If an item on this checklist is marked "incomplete," a reason will be provided. Items deemed "Not Applicable" are considered complete.

Exit Criteria for Design Testing:

Exit criteria can be summarized in the following points:

- All Design Test (system, integration, user acceptance) testing scripts and cycles have been executed successfully.
- All identified errors and defects, as well as issues and incidents, have been properly documented and worked through the resolution process.

- All reviews have been conducted and the reviews yielded satisfactory results.

2. Design Test Scope and Expectations

The FMS Phase IV Release (AR and AP) Design Test scope includes testing the following program interfaces:

- AP LARS Vendor Load Program
  - Initial load of LARS vendor records to FMS vendor tables
  - Addition of new records to LARS vendor tables
  - Modification of existing records in LARS vendor tables
  - Consistency of pre-existing records not meant to be updated
  - Deactivation / Reactivation of vendors
- Lender Customer Programs
  - Lender Customer Load Program
    - Initial load of customer information to AR tables
    - Addition of new LARS records to FMS customer tables
    - Modification to FMS customer records for existing customer updates
    - Consistency of pre-existing records not meant to be updated
  - Customer Vendor Xref Program
    - Addition of new records to Xref table
- AR / AP Invoices Interface
  - Delivery of 799 transactions to the AP module
  - Delivery of 799 transactions to the AR module
  - Creation of an AP invoice batch based on vendor data from LaRS
  - Creation of AR invoice batch based on customer data from LaRS
- Bank Lockbox Interface
  - Creation of AR invoices for transaction fee type of Sallie Mae
  - Creation of AR invoices for transaction fee type of Consolidation Loan Rebate Fee
  - Creation and application of receipt transaction for all imported fee types
  - Recognition of pre-existing receivable for transaction fee type of LARS (based upon sequence of oldest invoice, then interest, then principle)
- Stress testing for all above scopes

The Design Test cycles will include scenarios tested during unit testing and will include system / integration / user acceptance test scripts. In addition, the cycles will include the extensions, which are provided in the functional design documents for Phase IV Lender

Redesign efforts. The Integration Test cycle differs slightly as it involves interaction with LARS and ensures that coding introduced during Phase IV does not interfere with interfaces developed in Phase III (demonstrated through accounting expected results).

System, Integration, and User Acceptance test cycle planning and scheduling must be completed prior to the start date of the Design Test. This checkpoint has been included in the readiness checklist.

Sixteen weeks total have been allocated for the execution of the Design Tests and resolution of issues. The test cycle schedule is found in the Test Cycle Time Allocation Chart, located in the Appendix.

System, Integration, and User Acceptance testing are further divided into smaller cycles. These smaller cycles are defined in the System and Integration Test Cycle Breakdown Chart, located in the Appendix .

3. DT Test Schedule

Execution of Design Testing is tentatively scheduled for 06/03/2002 – 08/23/2002. A Test Readiness Review will be held prior to the start date of testing to verify all entry criteria have been met. The test execution phase involves executing each of the test cases and comparing the actual results with the expected results. Execution of the cases may sometimes occur more than once during the test execution phase, since discrepancies will be identified and fixed during this phase. Design Test cases will be executed until no discrepancies are found or other resolutions or work-arounds are identified. Additional details on the Design Test schedule can be found in the chart titled "Detailed Testing Time Table" located in the Appendix. This chart is subject to change at management discretion. All official dates will be reflected in the project plan.

4. DT Roles and Responsibilities

Staffing for DT involves both FSA FMS Development Team and FSA personnel. The primary roles for the Design Test effort include Test Coordinator(s), Program Functional Workstream, Application Development and Support (AD&S), Technical Infrastructure, Integration Workstream, Management office and others. A summary of roles and responsibilities is described below:

Test Coordinator

(Lynn Chang)

- Develop and maintain the Design Test plan and Design Test work products.
- Develop Design Test status reporting.
- Define issue resolution and recovery approach.
- Review test scripts inventory, scenario test cases and test data.
- Work with the program leads in coordinating test scripts to minimize cross-team testing conflicts.
- Develop, review and utilize Design Test readiness checklist to ensure needed application set up or configuration requirements are coordinated and performed.
- Coordinate all test activities for the team according to the test plan.
- Assist in evaluating and recording test results.
- Provide feedback and progress reports to management office regarding status of product test activities and significant issues.
- Assist in review of regression testing to validate changes to the application software and/or configuration. Regression testing will consist of testing the programs and affected sub-systems following the multi-org modification.

Program Workstream

(Lynn Chang, Cassie D'Agata, and additional functional team members as needed)

- Development of the System, Integration, User Acceptance test scripts.
- Execute the tests.
- Evaluate test results.

- Record all incidents and problems encountered during testing activities.
- Conduct regression testing to validate changes to the application software and / or configuration.

Management Office

(Cara Jonas, Jennifer Alden, Todd Elliott)

- Participate in the walkthrough of the Design Test Plan.
- Review all incidents and problems encountered during testing activities.
- Monitor the testing process to identify problems, mitigate potential risks and schedule slippage.
- Review test results with the test team to clarify questions, concerning system functionality and discrepancies with expected results, and ensure that the design testing activity was a valid and complete exercise.
- Confirm that the testing process is comprehensive in scope and is complete.
- Review and determine that Design Test Exit Criteria have been met.

Integration Workstream

(all FMS AR / AP Team members)

The integration workstream is comprised of a cross section of program team members, management office, and AD&S.

Technical Infrastructure

(FMS Technical Architecture Team)

The infrastructure team is responsible for the Instance Creation and preparation as well as System support during execution of the Design Test. The Design Test landscape should mirror the production landscape as much as possible with respect to architecture.

All configuration management procedures should be followed for all code changes and migrations to the test environment. All changes or updates to the testing environment will require change requests under configuration management guidelines.

5. Executing and Validating Test Cases

A test case or scenario is a discrete, executable activity that will return a predictable result. The scope of each test case will be identified on the description section of the test script file. Test cases are made up of a setup, test description, and expected result sections (see Exhibit 1). A test case is designed to verify the functionality of a business scenario, and describes step-by-step how the business scenario is being tested. Test scripts will address all of the requirements outlined in the requirements matrix for AP and AR and will identify what requirements are associated with each step.

Executing the test cases is the responsibility of the entire test team, as coordinated by the Test Team Coordinator. Any team member performing a test will update the appropriate fields of the test script matrix when conducting Design Testing.

Validation is defined to be the comparison of the actual results to the expected results. If the actual data results match the expected results then the test script passes. FSA CFO and FSA FMS personnel will be asked to initial test scripts that passed the controlled test script.

6. Issue Tracking and Problem Diagnosis and Resolution

To aid in issue tracking during Design Test, a Microsoft Access Database tool has been developed. This tool will contain all test cycle scripts, as well as link particular functions within the scripts to system requirements. The tool will also provide issue-tracking capability from problem diagnosis through resolution.

The test team member conducting the test will record any incidents or problems that are related to product testing. In order to provide an audit trail, updates to an incident will be tracked on the *original* incident report. An analysis of the incident will be performed to determine if the problem was caused by:

- Problem(s) with the test data;
- Problem(s) in the test environment;

- An incorrectly run test script;

- A misunderstanding of what the expected result should be.

If necessary, the incident will be forwarded to the business requirements, software architecture or technical support team for assistance in this analysis.

Incidents or problems will be recorded using the "Design Testing Incident Report" (see Exhibit 3). The "Design Testing Incident Report" is designed to capture as much information as possible to relay to the business requirements, software architecture or technical support team. This information will enable recreation of the situation. Test team members completing a "Design Testing Incident Report" should:

1. Write down a description of the symptoms that occurred and the test execution steps leading up to them, noting the date and time of the discrepancy.

2. Print any screen(s) or report(s) that illustrate the error that is occurring. Printouts are to be placed in the testing binder for IV&V review.

3. Escalate the problem to the Test Team Coordinator immediately if this is a "show stopper" that prevents continuation of the test.

4. If the problem isn't severe, log the information and continue with the same or other tests.

A "discrepancy" is a difference between what was *expected* to happen and what *actually* happened. Discrepancies are initially evaluated to ensure that they represent system defects. The initial evaluation of discrepancies includes:

- Review of the test data;

- Review of the test environment;

- Review of test cases;

- Review of actual system functioning.

If it is determined that the "discrepancy" represents a system defect, then the incident will be forwarded to the business requirements, software architecture or technical support team for assessment and proposed solution(s). The proposed solution(s) may

be technical / development or business process based (or both). Further, some solutions will entail significant technical or development work whereas others will not (and likewise for business process-based options). Once the solution options have been identified, the Test Team Coordinator will work with project management in determining which option will be followed or if the resolution will be deferred.

Standard procedures for development, unit test, code walkthrough and migration will be followed for any solution for which technical / development work is included. The solution will be tested as part of the Design Test, and regression testing as needed will be conducted for the systems and processes impacted by the modification. After the solution has been successfully tested, the incident will be designated as "closed" by the Test Team Coordinator.

7. Test Tracking and Documentation

The actual test results will be compared to the expected results to determine if the test ran correctly (or incorrectly). At the completion of each test activity the Test Team member will evaluate the status of the activity as "Pass", "Fail" or "Not Tested". A status of "Pass" indicates that the expected results were achieved. A status of "Not Tested" indicates that the the test step was not tested for reasons that will be described in the "Comments" field. A status of "Fail" indicates that actual results do not match expected results and troubleshooting will take place. Design Testing is considered complete when each test case has a status of "Pass", and the results have been reviewed by the Test Team Coordinator and the Management Office. All updates to status will be printed and placed in the Testing Binder for IV&V review.

8. Test Data

Sample data will be composed to complete Design Testing. The Testing Coordinator will work with the Lender Redesign team as well as Mellon when composing test data for Integration and User Acceptance Testing. All parties involved will work closely together to ensure that the integrity of the data is reflected as close as possible to true data. Test data used for each script will be printed out, where practical, and placed in the Testing Binder for IV&V review.

9. Test Scripts

A test script is a collection of related test cases, put into a sequence of steps. The Test Team Coordinator, the management office, and the system / integration test team will review all unit test and system / integration test scripts. All test scripts should be

placed under configuration management upon review by the Test Team Coordinator and management office.

10. Test Cycles

A test cycle is a collection of test scripts put into a sequence. A test cycle describes how a group of test scripts will be executed. Dependencies between scripts should be identified and recorded by FMS FSA functional program leads.

11. Design Test Results

All Design Test Results will be summarized and presented to management office for review and submission as a Design Test Exit Criteria checkpoint. The previously referred-to Microsoft Access tool will also be used to record Design Test Results in System and Integration Testing.

12. Client Acceptance Approach

The acceptance of Phase IV will be based on the review and acceptance of the Design Testing results. These results provide confirmation that the transactional data was processed into the current FMS instance and interfaced correctly. The FSA CFO and FSA CFO project Lead will provide production release approval contingent upon this acceptance.

13. EXHIBITS

Exhibit 1: System/Integration/User Acceptance Sample Test Script Template

*Note: Definitions of the fields contained within this script can be found in Exhibit 2.

Environment:			Script # / Name:					
Test Level:			Scenario Description:					
			File Name:					
Executed By / Date:								
Product / Release:								
Prepared By / Date:							Pass/Fail	
Acceptance Sign Off / Date								
Step	Action	Requirement #	Navigation Path	Input	Expected Results	Actual Results	Pass / Fail	Issues/Comments
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								

Exhibit 2 – Definitions of Test Script elements

Environment: Denotes environment in which script is run: DEVCOM, INTCOM, or INTPROD

Test Level: Denotes which phase of testing is being performed: Unit, System, Integration, or User Acceptance

Script Number/Name: Number in reference to the defined cycles and sub-cycles in Appendix 2, followed by the scenario name.

Scenario Description: Describe briefly the objective of the test, such as to verify that something works a certain way, or to ensure that invalid data is identified, etc.

File Name: Name of test file (containing appropriate test data) being run, if applicable

Executed by/Date: Person(s) executing the test and date of execution

Product/Release: Name and version of product being tested

Prepared by/Date: Script author and date of script creation

Acceptance Sign Off/Date: Signature of person executing (and passing) script, date

Pass/Fail: If script fails, Acceptance Sign Off is left blank.

Step #: Sequential number of test step

Action: Describe the anticipated test procedures or steps

Requirement #: Associated requirement number

Navigation Path: Detailed steps necessary to navigate through the program and complete the actions

Input: Input text for fields which require them

Expected Results: To be completed before the test is run. Describe the expected outcome of the test.

Actual Results: Describe whether the test resulted as expected or not; reference output documents, screen shots, etc., to provide documentation of results. This can be documented as a separate sheet.

Pass? Pass, Fail, or Not Tested

Issues/Comments: (as needed)

Exhibit 3 - Testing Incident Report

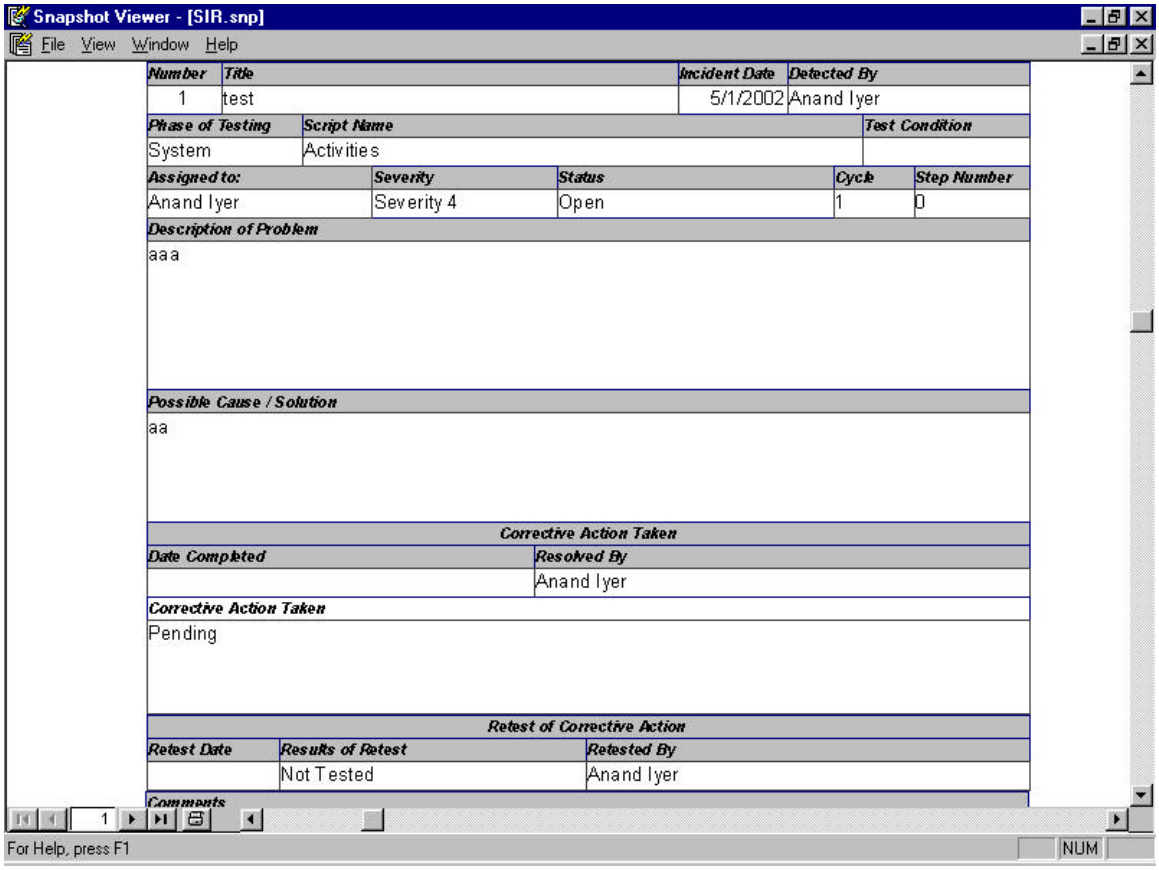

Exhibit 4. Readiness Check list

Phase IV Design Test Readiness Review Checklist

Workstream:

Deliverables	Owner(s)	Reviewer	Due Date	Signoff	Signoff Date	Notes/Document References
1) Design Documentation						
A. Design Documents Updated						
B. Requirements Matrix Updated						
C. Process Flows Completed						
2) Issues						
A. Phase IV Issues Resolved & Updated in FSA Tracker						
3) Unit Test Scripts						
A. Scripts Defined (incl. Data & Expected Results)						
B. Dependencies documentation						
C. Testers Assigned						
D. Data Requirements Identified						
E. Interface & Extension Execution Requirements Identified and Communicated to Tech Team						
F. Training Procedures & Navigations Linked to Tests						
4) System and Integration Test Scripts						
A. Test Cycle(s) and Scenario(s) Identified						
B. Test Flow(s) Developed						
C. Testers Assigned						
D. Data Requirements Identified						
5) Stress Test Requirements						
A. High-Frequency Process Identified						
B. Stress Test Scripts Created						
6) Configuration Guides						
A. Configurations Completed						
B. Module-specific Configurations Completed						
C. OFF Configurations Completed						
7) User Acceptance Tests						
A. Interface UATs completed						
B. Extension UATs completed						
C. Reports/Queries completed						
8) Baseline Instance						
A. Baseline Updated for Instance						
9) Responsibilities & Menus						
A. Custom Responsibilities & Menus Identified and created						
B. Participants Mapped to Responsibilities						
C. Information Communicated to Tech Team						
10) Pretesting						
A. Unit Test Scripts Pretested						
B. Responsibilities & Menus Pretested						
11) Instance						
A. Configurations Completed						
B. Manual Data Entered						
12) Training Documents						
A. Procedures Completed						
B. Navigations Completed						
13) Binders						
A. Workstream Binders Completed						
B. Binders Reviewed						

Certification

Based on a review of the items listed above, we have determined that:

☐ The Workstream is ready to begin the Design Test.

☐ The Workstream can begin the Design Test with the following conditions:

☐ The Workstream is not ready to begin the Design Test due to the following reasons:

Signoffs

FSA FMS Testing Coordinator Lynn Chang	
FSA FMS Transition Manager Christine Ponzi	
FSA FMS AR Project Lead Cara Jonas	
FSA FMS Implementation Lead Jennifer Alden	
IV&V Reviewer Phillip Norton	

14. APPENDICES

Appendix 1: Test Cycle Time Allocation

Test Cycle	Description	Scheduled Dates	Allocated Time
Unit Testing	Internal testing of AR and AP interfaces – performed by developers	5/06/2002 – 7/12/2002	20 days
System Testing	Internal testing of AR and AP interfaces – performed by functional workstream	6/03/2002 – 7/26/2002	40 days
Integration Testing	Incorporation of trading partners (LARS, Mellon)	7/15/2002 – 8/09/2002	20 days
User Acceptance Testing	Incorporation of potential users (FSA)	8/12/2002 – 8/23/2002	10 days

Note: Regression testing will be performed by the FMS test team.

Appendix 2: System, Integration, User Acceptance Test Cycle Breakdown

23-May

System	Cycle	Sub-cycle	Function*	System Test Script
AR	1. Customer Data Maintenance	a. Automated - Customer Load Program	Add New Customer	AR 1a
			Update Existing Customer	
			Inactivate Customer	
			Reactivate Customer	
		b. Manual	Manually Create a New Customer	AR 1b
			Manually Modify Customer	
			Manually Inactivate Customer	
			Manually Inactivate Customer Address	
			Manually Reactivate Customer	
	2. Billing and Invoice	a. Automated - AR Invoice Interface	Import AR Invoices	AR 2a
		b. Manual	Manually Create Transaction Batch	AR 2b
			Partial Invoice Write-Off	
			Invoice Write-off	
			Void a Transaction	
	3. Receipts and Application	a. Automated - AR Lockbox Interface	Create Receipts of All Types	AR 3a
			Apply LaRS Receipts to Open LaRS Receivables	
			All Other Types, Create Receivables and Apply Receipts	
		b. Manual	Manually Create Receipts	AR 3b
			Manually Apply Receipts	
			Manually Create Misc. Fee Transactions	
			Standard Reversal	
			Debit Memo Reversal	
	4. Dunning and Collections	a. Manual	Create Customer Call Action	AR 4a
			Update Customer Call Action	
			Assign/Accrue Finance Charges	
			Update Treasury Finance Charge Rate	
			Run Aging Report	
			Generate Dunning Letters	
			Run Receivable Notification	
	5. Month End Processes	a. Manual	Open/Close AR Accounting Periods	AR 5a
			Complete Manual Reconciliation	
			Manually Initiate AR Posting	
			Posting Reclass Extension	
			Run Monthly Activity Summary Report	
AP	1. Vendor Data Maintenance	a. Automated - Vendor Load Program	Add New Vendor	AP 1a
			Update Existing Vendor	
			Place Vendor on Hold	
			Remove Vendor Hold	
			Inactivate Vendor	
			Reactivate Vendor	
		b. Manual	Setup Vendor Bank	AP 1b
			Manually Enter New Vendor	
			Manually Update Vendor	
			Manually Inactivate Vendor	
			Manually Reactivate vendor	
			Run Suppliers Report	
	2. Invoices	a. Automated - AP Invoice Interface	Import AP Invoices	AP 2a
			Place a Hold on Invoice (Amount exceeding $10 M)	
			Review and Approve Invoices >$10M	
		b. Manual	Manually Create Invoice Batch	AP 2b
			Manually Hold Invoice	
			Approve Invoice Batch	
			Cancel Invoice	
			Release Hold	
			Run Invoice Register Report	
	3. Payments	a. Manual	Create Debit Memo in AP for Offset Transactions	AP 3a
			Manually Create Electronic Payment Batch	
			Manually Create Check Payment Batch	
			Force Payment of Invoice < $25	
			View Payment File	
			Run Payment Notification	
			Manually Void Payment	
			Create Credit Memo in AR for Offset Transactions	
	4. Treasury Confirmation	a. Manual	Confirm Payment Batch	AP 4a
	5. Month End Processes	a. Manual	Manually Initiate AP Posting	AP 5a
			Open/Close AP Accounting Periods	
			Run Monthly Activity Summary Report	
	6. Quarter End Processes	a. Manual	Run Quarterly Activity Summary Report	AP 6a

*Note: Each system test script may contain multiple (but related) functions. If one function fails, the entire script will fail. After debugging, scripts will be re-run until all functions pass.

Appendix 3: Detailed Testing Time Table

ID	Task Name	Start Date	End Date
1	Unit and DT Planning	04/29/02	05/10/02
2	Design Documentation	02/15/02	05/03/02
3	Setup/Configuration/Account Mapping	05/13/02	05/15/02*
4	Instance Creation and Preparation	05/09/02	05/10/02*
5	Pre-DT Reviews	05/06/02	05/31/02*
6	Testing Readiness Review	5/30/2002	5/30/2002
7	DT - System Testing	06/03/02	07/26/02*
8	DT – Integration Testing	07/15/02	08/09/02*
9	DT – User Acceptance Test	08/12/02	08/23/02*
10	Pre-Production Readiness Review (PRR)	08/26/02	08/26/02*
11	PRR	08/28/02	08/28/02*
12	Submission of Change Request	08/28/02	08/28/02*
13	Production Release	09/09/02	09/09/02*

*Note: Dates are subject to change. Updates will be made to this table.